WHAT IS CLAIMED IS:

1. A method, comprising:

monitoring paths between a first controller and second controller;

determining whether one path has been unavailable for a predetermined time period in response to detecting that the path is unavailable;

indicating the path in a first failed state if the path has been unavailable for more than the predetermined time period; and

indicating the path in a second failed state if the path has not been unavailable for the predetermined time period.

2. The method of claim 1, further comprising:

indicating the path in a functioning state if the path is determined to be available.

3. The method of claim 1, wherein the first failed state comprises a permanent failed state and the second failed state comprises a transient failed state.

4. The method of claim 1, further comprising:

receiving a write request;

returning fail to the write request in response to determining that all paths are in the first failed state; and

queuing the write request in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.

5. The method of claim 4, further comprising:

submitting the write request to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state.

6. The method of claim 4, wherein at least one primary volume managed by the primary controller and at least one secondary volume managed by the secondary

controller are designated as volume pairs, wherein writes to one primary volume in one volume pair is copied to the corresponding secondary volume in the volume pair, further comprising:

suspending one volume pair including the primary volume to which the write request is directed in response to determining that all paths are in the first failed state.

7. The method of claim 4, further comprising:

periodically processing the queue and write requests queued therein; and

submitting the write requests in the queue to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state

8. The method of claim 4, further comprising:

indicating a time the write request was received when queuing the write request in the queue; and

returning fail to one write request in the queue in response to determining that the write request has been queued longer than a request timeout period.

9. The method of claim 8, further comprising:

periodically processing the queue and write requests queued therein to determine whether to return fail to those write requests queued longer than the request timeout period.

10. The method of claim 9, further comprising:

determining whether at least one path is in a functioning state when periodically processing the queue, wherein fail is only returned to those write requests having been queued longer than the request timeout period in response to determining that no path is in the functioning state.

11. The method of claim 1, further comprising:

receiving a read request to access requested data;

returning the requested data with the first controller in response to determining that the data is available at a first storage coupled to the first controller;

determining that the requested data is not available at the first storage;

returning fail to the read request in response to determining that all paths are in the first failed state in response to determining that the data is not available at the first storage; and

queuing the read request in a queue to transfer to the secondary controller to access the requested data from a second storage in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state in response to determining that the data is not available at the first storage.

12. The method of claim 1, further comprising:

performing a failover to the second controller to service I/O requests through the second controller in response to detecting a failure related to the first controller;

logging updates made by the second controller during the failover;

transferring logged updates from the second controller to the primary controller in response to a failback to the first controller;

returning fail to the transfer of one logged update to the first controller in response to determining that all paths are in the first failed state; and

queuing one logged update to transfer to the first controller in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.

13. A system, comprising:

a first controller;

a second controller;

paths between the first and second controller;

code executed by the first controller to perform:

(i) monitoring paths between a first controller and second controller;

(ii) determining whether one path has been unavailable for a predetermined time period in response to detecting that the path is unavailable;

(iii) indicating the path in a first failed state if the path has been unavailable for more than the predetermined time period; and

(iv) indicating the path in a second failed state if the path has not been unavailable for the predetermined time period.

14. The system of claim 13, wherein the code is executed by the first controller to further perform:

indicating the path in a functioning state if the path is determined to be available.

15. The system of claim 13, wherein the first failed state comprises a permanent failed state and the second failed state comprises a transient failed state.

16. The system of claim 13, wherein the code is executed by the first controller to further perform:

receiving a write request;

returning fail to the write request in response to determining that all paths are in the first failed state; and

queuing the write request in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.

17. The system of claim 16, wherein the code is executed by the first controller to further perform:

submitting the write request to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state.

18. The system of claim 16, wherein at least one primary volume managed by the primary controller and at least one secondary volume managed by the secondary controller are designated as volume pairs, wherein writes to one primary volume in one volume pair is copied to the corresponding secondary volume in the volume pair, and wherein the code is executed by the first controller to further perform:

suspending one volume pair including the primary volume to which the write request is directed in response to determining that all paths are in the first failed state.

19. The system of claim 16, wherein the code is executed by the first controller to further perform:

periodically processing the queue and write requests queued therein; and

submitting the write requests in the queue to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state

20. The system of claim 16, wherein the code is executed by the first controller to further perform:

indicating a time the write request was received when queuing the write request in the queue; and

returning fail to one write request in the queue in response to determining that the write request has been queued longer than a request timeout period.

21. The system of claim 20, wherein the code is executed by the first controller to further perform:

periodically processing the queue and write requests queued therein to determine whether to return fail to those write requests queued longer than the request timeout period.

22. The system of claim 21, wherein the code is executed by the first controller to further perform:

determining whether at least one path is in a functioning state when periodically processing the queue, wherein fail is only returned to those write requests having been queued longer than the request timeout period in response to determining that no path is in the functioning state.

23. The system of claim 13, wherein the code is executed by the first controller to further perform:

receiving a read request to access requested data;

returning the requested data with the first controller in response to determining that the data is available at a first storage coupled to the first controller;

determining that the requested data is not available at the first storage;

returning fail to the read request in response to determining that all paths are in the first failed state in response to determining that the data is not available at the first storage; and

queuing the read request in a queue to transfer to the secondary controller to access the requested data from a second storage in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state in response to determining that the data is not available at the first storage.

24. The system of claim 13, further comprising:

code executed by the second controller to perform:

(i) performing a failover from the first controller to the second controller to service I/O requests through the second controller in response to detecting a failure related to the first controller;

(ii) logging updates during the failover;

(iii) transferring logged updates to the primary controller in response to a failback to the first controller;

(iv) returning fail to the transfer of one logged update to the first controller in response to determining that all paths are in the first failed state; and

(v) queuing one logged update to transfer to the first controller in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.

25. An article of manufacture for monitoring paths between a first controller and second controller, wherein the article of manufacture is capable of causing operations to be performed, the operations comprising:

determining whether one path has been unavailable for a predetermined time period in response to detecting that the path is unavailable;

indicating the path in a first failed state if the path has been unavailable for more than the predetermined time period; and

indicating the path in a second failed state if the path has not been unavailable for the predetermined time period.

26. The article of manufacture of claim 25, wherein the operations further comprise:

indicating the path in a functioning state if the path is determined to be available.

27. The article of manufacture of claim 25, wherein the first failed state comprises a permanent failed state and the second failed state comprises a transient failed state.

28. The article of manufacture of claim 25, wherein the operations further comprise:

receiving a write request;

returning fail to the write request in response to determining that all paths are in the first failed state; and

queuing the write request in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.

29. The article of manufacture of claim 28, wherein the operations further comprise:

submitting the write request to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state.

30. The article of manufacture of claim 28, wherein at least one primary volume managed by the primary controller and at least one secondary volume managed by the secondary controller are designated as volume pairs, wherein writes to one primary volume in one volume pair is copied to the corresponding secondary volume in the volume pair, wherein the operations further comprise:

suspending one volume pair including the primary volume to which the write request is directed in response to determining that all paths are in the first failed state.

31. The article of manufacture of claim 28, wherein the operations further comprise:

periodically processing the queue and write requests queued therein; and

submitting the write requests in the queue to one path indicated in the functioning state to transmit to the secondary controller in response to determining that at least one path is in the functioning state

32. The article of manufacture of claim 28,wherein the operations further comprise:

indicating a time the write request was received when queuing the write request in the queue; and

returning fail to one write request in the queue in response to determining that the write request has been queued longer than a request timeout period.

33. The article of manufacture of claim 32, wherein the operations further comprise:

periodically processing the queue and write requests queued therein to determine whether to return fail to those write requests queued longer than the request timeout period.

34. The article of manufacture of claim 33, wherein the operations further comprise:

determining whether at least one path is in a functioning state when periodically processing the queue, wherein fail is only returned to those write requests having been queued longer than the request timeout period in response to determining that no path is in the functioning state.

35. The article of manufacture of claim 25, wherein the operations further comprise:

receiving a read request to access requested data;

returning the requested data with the first controller in response to determining that the data is available at a first storage coupled to the first controller;

determining that the requested data is not available at the first storage;

returning fail to the read request in response to determining that all paths are in the first failed state in response to determining that the data is not available at the first storage; and

queuing the read request in a queue to transfer to the secondary controller to access the requested data from a second storage in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state in response to determining that the data is not available at the first storage.

36. The article of manufacture of claim 25, wherein the operations further comprise:

performing a failover to the second controller to service I/O requests through the second controller in response to detecting a failure related to the first controller;

logging updates made by the second controller during the failover;

transferring logged updates from the second controller to the primary controller in response to a failback to the first controller;

returning fail to the transfer of one logged update to the first controller in response to determining that all paths are in the first failed state; and

queuing one logged update to transfer to the first controller in a queue in response to determining that at least one path is in the second failed state and no paths are indicated in a functioning state.